Mining time-dependent communities

Qinna Wang, Eric Fleury

Mining time-dependent communities

Qinna Wang
D-NET/INRIA, LIP, Université de Lyon
Ecole Normale Supérieure de Lyon
Email: qinna.wang@ens-lyon.fr

Eric Fleury
D-NET/INRIA, LIP, Université de Lyon
Ecole Normale Supérieure de Lyon
Email: eric.fleury@inria.fr

***Abstract*—Time evolution is one important feature of communities in network science. It is related with capturing critical events, characterizing community members, and predicting behaviours of communities in networks with time varying. However, most of existing community detection techniques are proposed for static networks. Here, we present a new framework to uncover community structure for each temporal graph over time. In consideration of regularizing time-dependent communities, the high temporal variations will be prevented and the gained results on community evolution become more reasonable. Having applied it on synthetic networks, the experimental results offer new views in dynamic networks.**

***Index Terms*—dynamic network, community structure, overlapping community structure**

I. Introduction

The study on structural evolution is an important issue of science research. To our knowledge, it provides possibilities to capture critical events, characterize community members, and predict behaviours of communities in dynamic networks, for example, the discussions in Internet based systems (e.g. Facebook, Myspace, Twitter) suggest new topics and popular subjects; the ENRON email networks [1] show the position of employee and reveal the importances of employee during the crisis; And the CELLPHONE call communication networks [2] present the efficiency of cell phone call services.

Currently, many researchers proposed and developed diverse technologies [3], [4], [5] to capture community structure of networks. The traditional analysis treats the network as the static graph that is derived from the aggregation of interactions over time or is a particular temporal graph picked from the network dataset. Although it provided meaningful information in network sciences, many important characteristics or properties are ignored such as the temporal change of communities. In order to resolve this problem, we propose a new framework for mining time-dependent communities. The found communities correspond to community histories and the temporal graph. We use synthetic graphs to demonstrate its availability. We expect our studies be helpful in network science.

II. Related work

In [6], the community structure is supposed as the partition of networks, such that each node belongs to one and only one community, and the community structure corresponds to the partition with the maximum modularity value, or at least a very high value. Then, modularity optimization becomes one important technique for community detection, such as [7], [8] and [9]. However, it fails to resolve the problem of overlapping community structure, in which nodes are allowed to be shared between communities. There are also many resolutions, such as the CPM [10] which considers the communities as adjacent k-cliques (where adjacency means sharing $k-1$ nodes), the fitness optimization [11] which is derived from the definition of communities in the strong sense and the weak sense [12], and the link partition [13] which considers the problem of the node assignment as the link assignment.

Furthermore, the problem of dynamic community detection is another popular issue. Palla et al. [14] proposed a method to uncover communities by the CPM [10] and track communities by the relative overlap. Sun et al. [15] suggested GraphScope to mine the evolution of communities in using the MDL. And Mucha et al. [5] developed a multi-slice modularity to capture the community structure in time-dependent, multi-slices and multiplex networks.

In this paper, we introduce a new method to mine time-dependent overlapping communities. Rather than studying the structural properties without the consideration of community histories, our approach investigates the evolution of communities by regularizing communities. Moreover, overlapping community structure is considered to improve the accuracy of community mining results.

III. Community detection on static networks

Let us address the overlapping community detection in this section. Given a graph $\mathcal{G}(V, E)$, our method detects a set of cores, then expands these cores by optimizing a local community fitness function.

A. Notations and definitions

Fitness function A community fitness function reveals the edge density of a subgraph C, which is defined as:

$$f_C = \frac{\sum_{i\in C} k_i^{int}}{(\sum_{i\in C} k_i^{int} + \sum_{i\in C} k_i^{out})^{\alpha}} , \qquad (1)$$

where α is a tunable parameter, the internal degree k_i^{int} and external degree k_i^{out} are the number of edges from node i of C to other nodes inside of C or outside of C, respectively.

Previous studies [16] suggest the fitness function as a local optimization strategy for overlapping community detection. And its optimization technique can be summarized as:

1) For each node i which is adjacent to a community C, calculate its node fitness, i.e. $F_i = F_{C\cup i} - F_{C\setminus i}$.

2) Select the node i which brings the largest positive F_i.
3) If the node i exists, add it to C and repeat step 1; otherwise, stop and return the final community C.

Cores of communities The cores are embedded in communities. Our goal is expanding the cores by adding peripheral nodes (i.e. the nodes are not cores) to explore communities. We choose strong clusters (i.e. a set of nodes keeps stable memberships) as cores. Our choice is motivated by the observation that, overlapping nodes fail to keep stable membership in disjoint community detection. We assume that the probability p_{ij} of a pair of nodes (i, j) belonging to the same community implies memberships: *(i)* $p_{ij} \geq \beta$ represents (i, j) holding a stable membership ($\beta = 100\%$, typically); and *(ii)* if $(i, j) \in C$ and $(i, k) \in C$, then $(j, k) \in C$.

Under the assumption, a matrix $\mathbf{P} = [p_{ij}]_{n\times n}$ is applied to find strong clusters. Considering the results of an non-determined community detection algorithm depend on the ordering of clustering nodes, we obtain $\mathbf{P}$ by repeating Louvain algorithm [7] until $\|P_{ij}^{k+1} - P_{ij}^{k}\| < \varepsilon$, where P_{ij}^{k} represents the results after k runs.

Overlapping size The overlapping size R is proposed to identify communities at different time steps a and b in [14]. For two communities $C_{i;a}$ and $C_{j;b}$, their overlapping size is:

$$R_{C_{j;b}}(C_{i;a}) = \frac{|C_{i;a} \cap C_{j;b}|}{|C_{i;a} \cup C_{j;b}|} \,. \tag{2}$$

Among the founded partitions of $\mathcal{G}$, we select the partition $\mathcal{P}_{max}$ which has the maximum modularity to choose cores. For a community $C_i \in \mathcal{P}_{max}$, we select the strong cluster C' which has the largest overlapping size $R_{C'}(C_i)$ to its core.

Extended modularity The modularity which is proposed by Newman et al. [6] to find and evaluate the community structure has several modifications, such as [17]:

$$Q(\mathcal{C}) = \sum_{i=1}^{c} \left[\frac{k_{C_i}^{int}}{2m} - \left(\frac{k_{C_i}^{tot}}{2m} \right)^2 \right] \,, \tag{3}$$

where $k_{C_i}^{int}$ is the sum of weights of edges inside C_i, and $k_{C_i}^{tot}$ is the sum of weights of edges incident the nodes inside C_i. We use the extend modularity to quantify the found community structure and select the tunable α (see Eq. 1) corresponding the maximum extended modularity value.

B. Overview of static community detection

We have reviewed the fitness function, the cores of communities, the overlapping size and the extended modularity function. Then we outline our method in the following:

1) Find strong clusters C' by repeating Louvain algorithm until the convergence of $\mathbf{P}$.
2) Select one strong clusters C' which has the largest overlapping size with the community $C_i \in \mathcal{P}_{max}$ to the core for the expansion until no addition of peripheral nodes would improve its fitness.
3) Go back and continue step 2 until all cores have been expanded.

Supposing the Louvain algorithm [7] is applied, the running time of our method is in complexity $\mathcal{O}(K|E| + M|E|)$, where K is the number of running Louvain algorithm to gain the matrix $\mathbf{P}$, M is the number of cores to be expanded.

IV. Our method

In this section, we describe our framework in detail. At a given time t, we denote the temporal graph by $\mathcal{G}_t$, the community structure by $\mathcal{C}_t = \{C_{1;t}, \ldots, C_{c;t}\}$, and the set of cores by $\mathcal{C}_{s;t} = \{C_{s1;t}, \ldots, C_{sc;t}\}$. The principle is that the community structure $\mathcal{C}_t$ should follow certain structural organizations of $\mathcal{G}_{t-1}$. It makes the community evolution more regular and reasonable.

We mentioned the cores and the ordering of clustering nodes in Sec. III. Initializing each core $C_{si;t-1} \in \mathcal{C}_{s;t-1}$ into a sub-community before community detection on $\mathcal{G}_t$, the community structure $\mathcal{C}_t$ is naturally related with its history. To make the community mining results more reasonable, we randomly select the portion $r \in [0, 1]$ of the core nodes $i \in \mathcal{C}_{s;t-1}$ into a single sub-community when running Louvain algorithm for the matrix $\mathbf{P}_t$ calculation. Our modification makes our results be affected with the parameter r.

Furthermore, the problem of how to track the evolution of communities is critical in characterizing networks with time varying. Motivated by Greene et al. [18], we select the set of fronts $\mathcal{F} = \{F_1, \ldots, F_k\}$ to denote dynamic communities $\mathcal{D} = \{D_1, \ldots, D_k\}$, where each F_i is the first core of community D_i (i.e. when a community $C_{i;t}$ is found and it fails to match any dynamic community, we consider a new dynamic community D_i is created whose front is $C_{si;t}$, where $C_{si;t}$ is the core of $C_{i;t}$). The matching between $C_{si;t}$ and D_j is through the overlapping size $R_{C_{i;t}}(F_j)$ with a matching threshold $\theta \in [0, 1]$: if $R_{C_{si;t}}(F_j) \geq \theta$, $C_{si;t}$ is considered as the observation of D_j at time t; if several fronts fit, $C_{si;t}$ is matched in descending ordering of their overlapping size; otherwise, a new dynamic community is formed and denoted by the front $F_{k+1} \equiv C_{si;t}$.

V. Experimental Studies

A. Benchmark graphs

To validity our method, we apply it on the benchmark graphs, which are proposed by Greene et al. [18]. These graphs are constructed by embedding nodes into communities [11]: the edges are randomly assigned according to the ground truth and a set of parameters (e.g. the mixing parameter is 0.2, the average degree is 20, the maximum degree is 40), while the evolutions of communities are controlled by the community events: intermittent communities, in which 10% of existing communities are unobserved for their concealment at each time step; expansion and contraction, in which 40 randomly selected communities expand or contract their 25% size at each time step; birth and death, in which 40 new communities are constructed to replace 40 existing communities; merging and splitting, in which 40 communities are randomly selected to be split, while 40 cases of the merging of two random communities happen. Applying the parametric modularity [19] with the resolution parameter $\gamma = 58$, we obtain the community

structure having the similar number of communities as the ground truth.

Figure 1 shows the comparison between the found community structure and the ground truth in terms of NMI [11] for a range of tunable parameter $r \in [0.01, 0.8]$. The more the result matches the ground truth, the higher the NMI is. If the found result totally matches the ground truth, NMI is 1. We observe that our method has the high accuracy in community detection; the difference of NMI among results is few for different parameter values r; and the proportionality $r = 0.50$ has a better performance in the cases of merging and splitting. Generally, our method has good performances in community detection on dynamic networks.

B. The blog dataset

Next, we apply the dynamic blog dataset which involves into the post among blogs and is gained by aggregating interactions among blogs. Therefore, it's a growing pattern network.

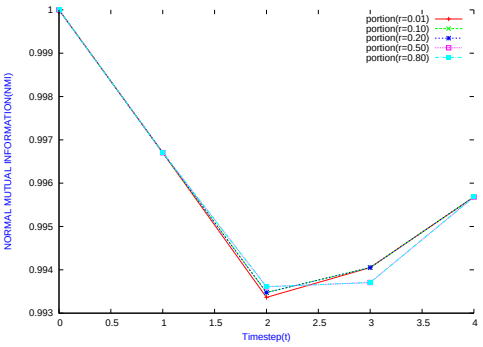
For simplicity, we select the modularity (Eq.3) for community detection, and only show the evolution of core nodes during $\{\mathcal{G}_1, \ldots, \mathcal{G}_9\}$ and $\{\mathcal{G}_{81}, \ldots, \mathcal{G}_{89}\}$ for $r = 0.01, r = 0.50$ (with $\theta = 0.10$, see Sec. IV), where the color shows the assignment of core nodes and the black denotes the absent of the node or that it is not the core node at time t.

Comparing Fig. 2 and Fig. 3, we note that the evolution of core nodes in $\{\mathcal{G}_1, \ldots, \mathcal{G}_9\}$ is less stable than $\{\mathcal{G}_{81}, \ldots, \mathcal{G}_{89}\}$: more nodes are added into the core set, more parts of core nodes change their memberships, and more emergence of new dynamic communities over time in Fig. 2. It reveals that the community structure is easily affected by the temporal variation if the test network is not gained with certain aggregations of interaction information.

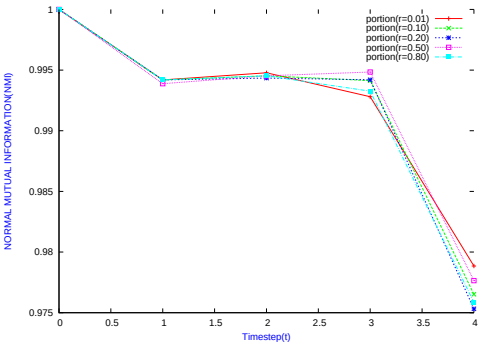
Considering the outputs for different values $r = 0.01, r = 0.50$, we observe the few difference: the similar number of dynamic communities, the similar core node assignment, the similar modularity value and the similar tendency of modularity (see Fig. 4). It seems that the outputs of our method is little influenced by the proportionality r. To prove it, we select the core node set at time $t = 81$ for $r = 0.01$ as the front set $\mathcal{F}$ (see Sec. IV) to identify communities for $r = 0.50$ (see Fig. 3). Then nearly all communities for $r = 0.01$ during $\{\mathcal{G}_{81}, \ldots, \mathcal{G}_{89}\}$ can be observed in the community structure for $r = 0.50$ (only 3 among 28 for $r = 0.50$ are failed to be identified), where results in Fig. 3 are gained following the historic evolution since $\mathcal{G}_1$.

VI. Conclusion

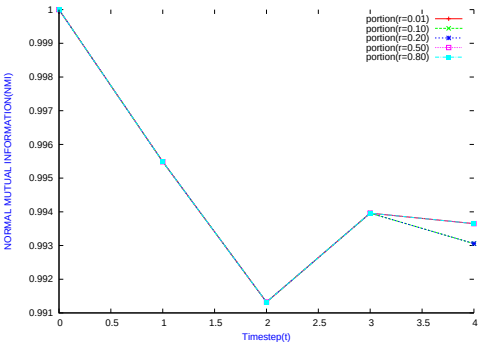
In this paper, we introduce a new framework to resolve the problem of dynamic community detection, which regularizes communities by initializing sub-communities corresponding the cores and the proportionality r. To prove its availability, we apply it on synthetic networks, and gain perfect experimental performances. The outputs of our method on the blog dataset show the results at the beginning of the time steps for such an aggregation network are easily influenced by the temporal variation. Additionally, the parameter value r has slight effects



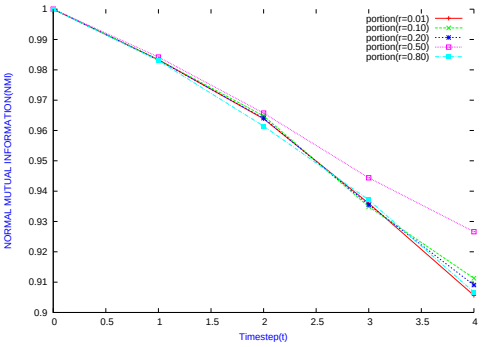
(a) Intermittent communities events



(b) Expansion and contraction events

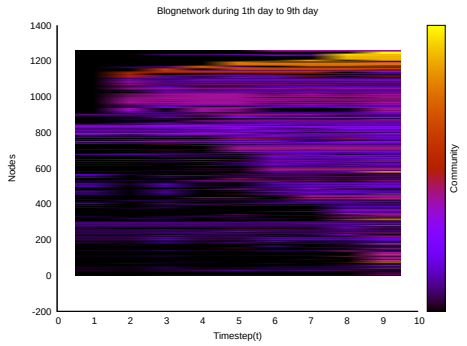


(c) Birth and death events

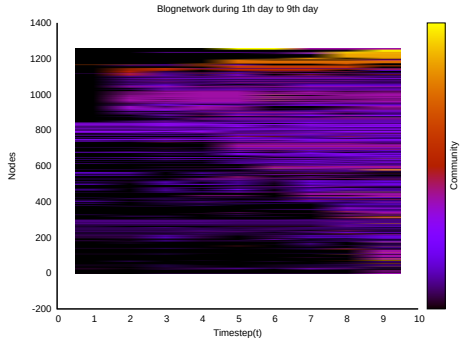


(d) Merging and splitting events

Fig. 1. In terms of the NMI, performances of our method on four synthetic graphs containing different types of evolution events with time variations.

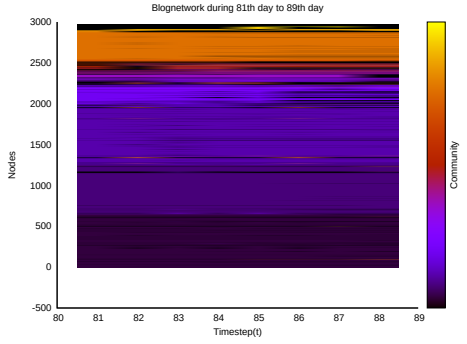


(a) Evolution of core nodes for $r = 0.01$

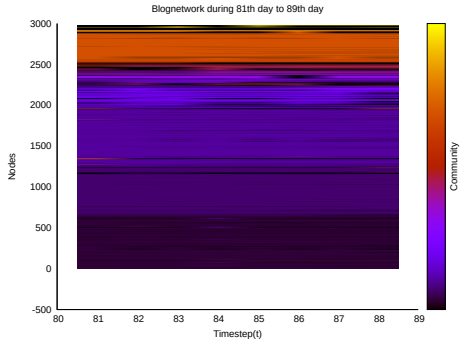


(b) Evolution of core nodes for $r = 0.50$

Fig. 2. Results of our algorithm on the blog dataset during 1st to 9th days for $r = 0.01, r = 0.50$. Colors depict the assignment of nodes and nodes are in the same order in Fig. 2(a) and Fig. 2(b).



(a) Evolution of core nodes for $r = 0.01$



(b) Evolution of core nodes for $r = 0.50$

Fig. 3. Results of our algorithm on the blog dataset during 81st to 89th days for $r = 0.01, r = 0.50$. Colors depict the assignment of nodes and nodes are in the same order in Fig. 3(a) and Fig. 3(b).

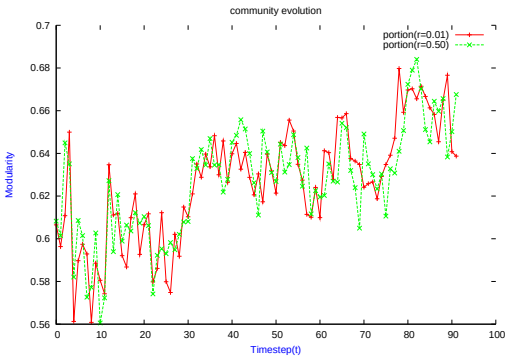


Fig. 4. Modularity on the blog dataset over time for $r = 0.01, r = 0.50$.

in the outputs. We expect our method be a powerful tool in mining features and properties of complex networks.

ACKNOWLEDGEMENTS

We thank WebFluence ANR to provide the blog dataset.

REFERENCES

[1] J. Diesner and K. M. Carley, "Exploration of communication networks from the enron email corpus," in *Proc. of Workshop on Link Analysis, Counterterrorism and Security, SIAM-SDM*, 2005, pp. 21–23.
[2] T. A. Taber, P. A. Alberto, A. Seltzer, and M. Hughes, "Obtaining assistance when lost in the community using cell phones," *Research and Practice for Persons with Severe Disabilities*, pp. 105–116, 2003.
[3] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *KDD '06*.
[4] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 2, pp. 1–31, 2009.
[5] P. J. Mucha, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 329, no. 5989, pp. 277–277, 2010.
[6] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E 69, 026113*, 2004.
[7] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical*, 2008.
[8] Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev.E 74, 036104*, 2006.
[9] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, no. 4, 2008.
[10] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
[11] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *ArXiv e-prints*, 2009.
[12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *PNAS*, vol. 101, no. 9, pp. 2658–2663, 2004.
[13] T. S. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, 2009.
[14] G. Palla, A. L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.
[15] J. M. Sun, P. S. Yu, S. Papadimitriou, and C. Faloutsos, "Graphscope: Parameter-free mining of large time-evolving graphs," *Kdd-2007*.
[16] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," *ArXiv e-prints*, Feb. 2010.
[17] S. W. And, "A spectral clustering approach to finding communities in graphs," in *SDM*, 2005, pp. 43–55.
[18] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," *ASONAM*, 2010.
[19] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, no. 1, p. 016110, Jul 2006.